Message Authentication Codes

CS 276: Introduction to Cryptography

Sanjam Garg

February 24, 2026

Overview

1. Message Authentication Codes
2. Fixed-Length MAC from PRF
3. Variable-Length MACs

MACs: Integrity and Authenticity

Goal

Bob wants to verify that the message came from Alice and was not tampered with. A **Message Authentication Code (MAC)** produces a tag t for m that cannot be forged.

MACs: Integrity and Authenticity

Goal

Bob wants to verify that the message came from Alice and was not tampered with. A **Message Authentication Code (MAC)** produces a tag t for m that cannot be forged.

Usage

Alice sends (m, t). Bob runs Verify(k, m, t); accepts if 1, rejects if 0.

MAC Scheme

Definition 1 (MAC Scheme)

A **MAC scheme** Π is a tuple $(\mathsf{Gen}, \mathsf{Mac}, \mathsf{Verify})$:

1. $k \leftarrow \mathsf{Gen}(1^n)$
2. $t \leftarrow \mathsf{Mac}(k, m)$
3. $\mathsf{Verify}(k, m, t) \in \{0, 1\}$

where n is the security parameter and $k, m, t \in \{0, 1\}^*$.

MAC Correctness

Definition 2 ((Perfect) MAC Correctness)

$\Pi = (\mathsf{Gen}, \mathsf{Mac}, \mathsf{Verify})$ is **correct** if for all n, all k output by $\mathsf{Gen}(1^n)$, and all $m \in \{0,1\}^*$:

$$\Pr[\mathsf{Verify}(k, m, \mathsf{Mac}(k, m)) = 1] = 1$$

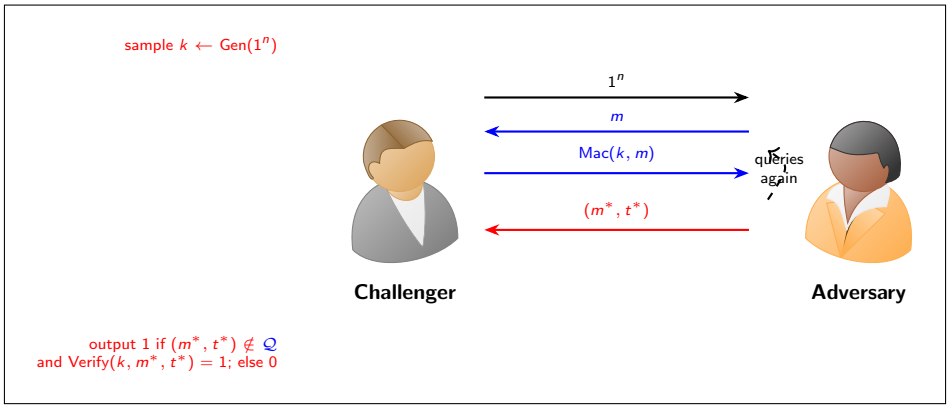
EUF-CMA Security

Game MAC-forge$_{\mathcal{A},\Pi}(n)$

1. Challenger samples $k \leftarrow \mathrm{Gen}(1^n)$; $\mathcal{A}$ gets 1^n
2. **Query**: $\mathcal{A}$ submits $m^{(i)}$; gets $t^{(i)} \leftarrow \mathrm{Mac}(k, m^{(i)})$. Let $\mathcal{Q} = \{(m^{(1)}, t^{(1)}), \ldots, (m^{(q)}, t^{(q)})\}$
3. **Forgery**: $\mathcal{A}$ outputs (m^*, t^*). Output 1 if $(m^*, t^*) \notin \mathcal{Q}$ and $\mathrm{Verify}(k, m^*, t^*) = 1$; else output 0.

MAC-forge: Challenger vs Adversary

$\text{MAC-forge}_{\mathcal{A},\Pi}(n)$



EUF-CMA Security

Definition 3 (EUF-CMA Security)

Π is **EUF-CMA-secure** if for all non-uniform PPT $\mathcal{A}$:

$$\Pr[\text{MAC-forge}_{\mathcal{A},\Pi}(n) = 1] = \text{negl}(n)$$

Fixed-Length MAC from PRF

Theorem 4

If $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is a secure PRF, then the following MAC for length n is EUF-CMA-secure:

- $\mathsf{Gen}(1^n)$: *output* $k \xleftarrow{\$} \{0,1\}^n$
- $\mathsf{Mac}(k, m)$: *output* $t = F_k(m)$
- $\mathsf{Verify}(k, m, t)$: *return* 1 *iff* $t = F_k(m)$

Fixed-Length MAC from PRF

Theorem 4

If $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ *is a secure PRF, then the following MAC for length* n *is EUF-CMA-secure:*

- $\mathrm{Gen}(1^n)$*: output* $k \xleftarrow{\$} \{0,1\}^n$
- $\mathrm{Mac}(k, m)$*: output* $t = F_k(m)$
- $\mathrm{Verify}(k, m, t)$*: return* 1 *iff* $t = F_k(m)$

Proof idea

Forger $\mathcal{A}$ gets tags for queries $m^{(i)}$, outputs (m^*, t^*). Adversary $\mathcal{B}$ against PRF forwards $\mathcal{A}$'s queries to F-oracle, then checks $F(m^*) \stackrel{?}{=} t^*$. If F is random, forgery probability $\leq 1/2^n$; if F is PRF, $\mathcal{A}$ forges with non-negligible probability. So $\mathcal{B}$ distinguishes.

Fixed-Length MAC: Reduction

Reduction $\mathcal{B}$

- $\mathcal{B}$ gets oracle $F(\cdot)$ (either F_k or random R_n).
- On $\mathcal{A}$'s query $m^{(i)}$: forward to F, return $t^{(i)}$ to $\mathcal{A}$.
- When $\mathcal{A}$ outputs (m^*, t^*): query m^*, get t; output 1 iff $t = t^*$.

Fixed-Length MAC: Reduction

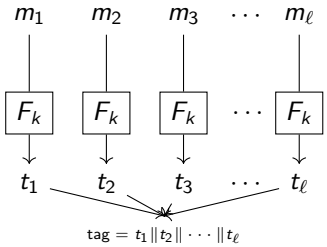
Reduction $\mathcal{B}$

- $\mathcal{B}$ gets oracle $F(\cdot)$ (either F_k or random R_n).
- On $\mathcal{A}$'s query $m^{(i)}$: forward to F, return $t^{(i)}$ to $\mathcal{A}$.
- When $\mathcal{A}$ outputs (m^*, t^*): query m^*, get t; output 1 iff $t = t^*$.

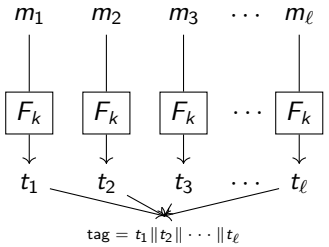
Analysis

$|\Pr[\mathcal{B}^{F_k(\cdot)}(1^n) = 1] - \Pr[\mathcal{B}^{R_n(\cdot)}(1^n) = 1]| = |\varepsilon_{\mathcal{A}}(n) - 1/2^n|$. If $\mathcal{A}$ forges with non-negligible $\varepsilon_{\mathcal{A}}$, then $\mathcal{B}$ distinguishes (random gives forgery prob $\leq 1/2^n$).

Variable-Length MACs: Concatenation of Tags



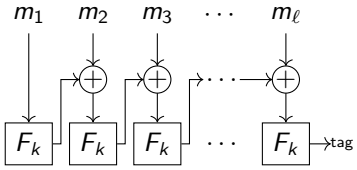
Variable-Length MACs: Concatenation of Tags



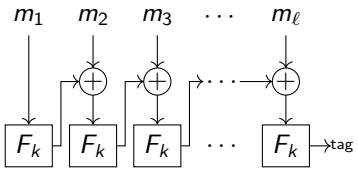
Attack

Query tag for 0^n (one block); get $t_1 = F_k(0^n)$. For $0^{2n} = 0^n \| 0^n$, the tag is $F_k(0^n) \| F_k(0^n) = t_1 \| t_1$. Adversary forges $(0^{2n}, t_1 \| t_1)$ without querying 0^{2n}. **Not secure.**

Variable-Length MACs: Chaining



Variable-Length MACs: Chaining



Length-extension attack

Query tag t for 0^n. Then $t' = F_k(0^n \oplus t)$ is the tag for 0^{2n}. Adversary gets valid tag for a new message without querying it. **Not secure.**

Expensive

One PRF call per block.

Variable-Length MACs: Carter–Wegman

Better approach

Use a **universal hash** $h_s(m)$ then $F_k(h_s(m))$ (Carter–Wegman). **One PRF call per message**.

Variable-Length MACs: Carter–Wegman

Better approach

Use a **universal hash** $h_s(m)$ then $F_k(h_s(m))$ (Carter–Wegman). **One PRF call per message**.

Why it works

Preprocess message with h_s; then a single F_k on the hash output. Collisions in h_s are unlikely (universal); if no collision, security reduces to fixed-length MAC.

Universal Hash Function

Definition 5 (Universal Hash)

$h : \mathcal{K} \times \mathcal{X}^* \to \mathcal{F}$ (with $\mathcal{F}$ field of size 2^m) is **universal** if for all distinct m, m' of length $\leq \ell$:

$$\Pr_s[h(s, m) = h(s, m')] \leq \frac{\ell}{|\mathcal{F}|}$$

Important

We **fix** m, m' and **sample** s. (If we fix s, we can find $m \neq m'$ that collide.)

Universal Hash: Polynomial Example

Construction

$h(s, m_0, \ldots, m_{\ell-1}) = m_0 + m_1 s + \cdots + m_{\ell-1} s^{\ell-1} + s^{\ell}$ in $\mathbb{F}_{2^m}$.

Claim: h is universal

For fixed distinct m, m', consider $h(s, m) - h(s, m')$. This is a nonzero polynomial in s of degree $\leq \ell$, so it has at most ℓ roots. Hence $\Pr_s[h(s, m) = h(s, m')] \leq \ell / |\mathcal{F}|$.

Carter–Wegman MAC: Security

MAC

$\mathrm{Mac}(k, s, m) = F_k(h(s, m))$ with universal h. Secure (Carter–Wegman style).

Carter–Wegman MAC: Security

MAC

$\mathrm{Mac}(k, s, m) = F_k(h(s, m))$ with universal h. Secure (Carter–Wegman style).

Proof idea

- $\mathcal{B}$ samples s, forwards $h_s(m)$ to F-oracle, returns tags to $\mathcal{A}$. If $\mathcal{A}$ forges (m^*, t^*), $\mathcal{B}$ queries $h_s(m^*)$, checks $t = t^*$.
- Let $E =$ "collision in h_s among $\mathcal{A}$'s queries and m^*". If $\neg E$, same as fixed-length MAC (random distinct inputs).
- If $\Pr(E)$ non-negligible: build $\mathcal{B}'$ that answers $\mathcal{A}$'s MAC queries with **random** responses; picks random $i < j$ (interpreted as the **first** pair of queries where a collision $h_s(m_i) = h_s(m_j)$ occurs), outputs (m_i, m_j). Collision with prob $\approx \Pr(E)/q^2$; breaks universality.

MAC for Arbitrary Length

Problem with padding

Padding with 0s: m and $m \| 0^n$ get the same tag. Insecure.

MAC for Arbitrary Length

Problem with padding

Padding with 0s: m and $m \| 0^n$ get the same tag. Insecure.

Solution

Put the **length** of the message in the first block; padding at the end. If messages differ by length, first block differs; if same length, padding is ignored. Gives secure MAC for arbitrary-length messages (universal hash + PRF).